Amendments to the Claims

Sub B1

Please amend the claims as follows:

1. (Currently amended): A method of performing multiple operations on a memory device, comprising:

dividing the memory device into k partitions, wherein k is an integer greater than or equal to two;

performing code operations from m code partitions out of k total partitions, wherein m is an integer greater than or equal to one; and

performing data operations from n data partitions out of k total partitions through low level functions accessed from the code partitions at approximately the same time as the code operations are performed from the m code partitions, wherein n is an integer greater than or equal to one.

2. (Original): The method of claim 1, wherein the data partitions and the code partitions do not overlap each other in the memory device.

3. (Original): The method of claim 1, wherein the m code partitions and the n data partitions equal the k total partitions.

4. (Original): The method of claim 3, wherein each of the m code partitions are equal in size to each of the n data partitions.

5. (Original): The method of claim 3, wherein the m code partitions and the n data partitions are fixed in memory space.

6. (Original): The method of claim 1, wherein the memory device is a flash memory.

7. (Original): The method of claim 6, wherein the flash memory is a flash

electrically erasable read only memory (EEPROM) array.

8. (Original): An apparatus comprising:

means for partitioning a memory device to enable multiple operations to be performed on a memory device at the same time; and

means for tracking operations performed on the device to restore interrupted tasks.

9. (Original): The apparatus of claim 8, further comprising a means for saving a preempted operation before entering an interrupt routine.

10. (Original): The apparatus of claim 8, further comprising a means for restoring a preempted task following an interrupt routine.

11. (Original): A memory array, comprising:

a plurality of partitions;

a status mode to provide partition status from the memory device;

a read mode to read code and data from the memory device; and

a write mode to write data to the memory device.

12. (Original): The memory array of claim 11, wherein the code is programmed into the memory array.

13. (Original): The memory array of claim 11, wherein the write mode is also capable of performing erase operations on data stored in the memory array.

14. (Original): The memory array of claim 11, wherein the memory array is a flash memory array.

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. (Cancelled)

19. (Original): An apparatus, comprising;

a memory device with k partitions, wherein k is an integer greater than or equal to two;

low level functions to access the memory device; and

a flag to indicate when a suspend operation has occurred.

20. (Original): The apparatus of claim 19, wherein the memory device comprises:

m code partitions, wherein m is an integer greater than or equal to one; and

n data partitions, wherein n is an integer greater than or equal to one.

21. (Original): The apparatus of claim 19, wherein the memory device is a flash memory.

22. (Cancelled)

23. (Cancelled)

24. (Cancelled)

25. (New) A method, comprising:

running a first operation on a first partition of a memory array;

running a first operation on a second partition of the memory array;

requesting a second operation to be performed on the first partition; and

determining if the second operation has a higher priority than the first operation.

26. (New) The method of claim 25, further comprising:

suspending the first operation if the second operation has a higher priority

than the first operation.

27. (New) The method of claim 26, further comprising:

setting a flag to indicate that the first operation must resume after the second operation is completed.

28. (New) The method of claim 26, further comprising:

running the second operation in the first partition.

29. (New) The method of claim 25, further comprising:

ignoring the request to perform the second operation if the first operation has a higher priority than the second operation.